

Přepracování řídicího programu malých průmyslových žehliček.

Firmware Redesigning of Small Flatwork Ironers.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2009

.....

Rád bych na tomto místě poděkoval ing. Petru Olivkovi a ing. Mojmíru Kotáskovi za cenné rady při zpracování této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá přepracováním zdrojového kódu řídicího programu malých průmyslových žehliček tak, aby splňoval firemní požadavky a pravidla pro strukturovanost a přehlednost kódu. Stávající zdrojový kód je podroben analýze, logika ovládání a funkce stroje je přepracována do formy stavových automatů, navržena a provedena je modularizace zdrojového kódu a doplnění nových funkcí a vlastností žehličky. Veškeré úpravy jsou provedeny s ohledem na budoucí přechod na novou hardwarovou platformu s možností běhu řídicího programu na real-time operačním systému.

Klíčová slova: průmyslový žehlič, řídicí software, C, C++

Abstract

This diploma thesis is about redesign of source code of small ironer's firmware to follow company's requirements and rules of source code structuralization and clearness. Current source code is analyzed, control logic and machine's functionality is redesigned into form of state automats, source code modularization is designed and implemented and new functionalities of ironer are added. All changes are made with respect to future migration to new hardware platform with possibility to run firmware on real-time operating system.

Keywords: industrial ironer, firmware, C, C++

Seznam použitých zkratk a symbolů

COIN	– Verze průmyslového žehliče vybavená mincovníkem nebo centrálním platebním systémem, určená pro neškolenou obsluhu ve veřejných prádelnách
EEPROM	– Electrically Erasable Programmable Read-Only Memory
GUI	– Grafický uživatelský interface
GPL	– GNU GENERAL PUBLIC LICENSE
IDE	– Integrated Development Enviroment - integrované vývojové prostředí
LED	– Light-emitting diode
OPL	– Verze průmyslového žehliče určená pro školenou obsluhu
RT systém	– Real Time systém - operační systém s během aplikací v reálném čase

Obsah

1	Úvod	9
1.1	Princip funkce žehliče	9
1.2	Konstrukční varianty žehliče	9
1.3	Řídící jednotka žehliče	9
1.4	Klávesnice řídící jednotky	10
1.5	Princip funkce skládání IF varianty žehliče	14
2	Analýza stávajícího zdrojového kódu	15
2.1	Zjištěné nedostatky stávajícího zdrojového kódu řídícího programu	19
3	Přeprogramování řídicí logiky ovládání stroje	21
3.1	Obsluha hlavní smyčky řízení stroje	21
3.2	Obsluha hlavního stavového automatu	21
3.3	Obsluha stavového automatu inicializace řídícího programu	21
3.4	Obsluha stavového automatu editace konfiguračních dat	22
3.5	Obsluha stavového automatu diagnostiky stroje	24
3.6	Obsluha stavového automatu běhu stroje	24
3.7	Obsluha stavového automatu mincovníku	25
3.8	Obsluha stavového automatu zvuku	25
4	Modularizace řídícího programu s ohledem na hardwarové závislosti software	35
4.1	Moduly složky <i>func</i>	35
4.2	Moduly složky <i>hw</i>	36
5	Rozšíření a doplnění stávajících funkcí řídícího programu	39
5.1	Návrh a implementace datové struktury pro uchování konfiguračních dat	39
5.2	Teplotní hystereze	41
5.3	Volba verze OPL nebo COIN za běhu stroje	41
6	GUI nádstavba řídícího programu	43
6.1	Návrh GUI nádstavby řídícího programu	43
6.2	Volba vhodného IDE	43
7	Možnosti portace na RT systém	47
8	Závěr	49
9	Reference	51
	Přílohy	51
A	Schémata a diagramy	53

Seznam tabulek

1	Tabulka konstrukčních variant průmyslového žehliče	10
2	Seznam a popis logických vstupů řídicí jednotky	11
3	Seznam a popis logických výstupů řídicí jednotky	11
4	Seznam a popis tlačítek klávesnice řídicí jednotky	13
5	Popis stavů hlavního stavového automatu	21
6	Popis stavů stavového automatu inicializace řídicího programu	22
7	Popis stavů stavového automatu editace konfiguračních dat	23
8	Popis stavů stavového automatu diagnostiky stroje	24
9	Popis stavů stavového automatu běhu stroje	25
10	Popis stavů stavového automatu mincovníku	25

Seznam obrázků

1	Vzhled klávesnice žehliče pro funkční variantu IF	12
2	Vývojový diagram stávajícího řídicího programu	16
3	Vývojový diagram hlavní běhové smyčky řídicího programu	17
4	Vývojový diagram funkcí stávajícího řídicího programu závislých na časovači 100 ms	18
5	Vývojový diagram hlavní funkce <i>main</i> přepracovaného řídicího programu	22
6	Vývojový diagram hlavní prováděcí funkce přepracovaného řídicího programu - část 1	27
7	Vývojový diagram hlavní prováděcí funkce přepracovaného řídicího programu - část 2	28
8	Hlavní stavový automat přepracovaného řídicího programu	29
9	Stavový automat inicializace řídicího programu	30
10	Stavový automat editace konfiguračních dat	31
11	Stavový automat diagnostiky stroje	32
12	Stavový automat běhu stroje	32
13	Stavový automat mincovníku	33
14	Vzhled GUI nastavy řídicího programu na platformě Microsoft Windows	44
15	IDE Code::Blocks na platformě Microsoft Windows	46
16	Schematicky znázorněný princip funkce žehliče	54
17	Schematicky znázorněný princip funkce skládání prádla	55

Seznam výpisů zdrojového kódu

1	Datová struktura PORT pro ukládání logických stavů a příznaků	20
2	Příklad oddělení hardwarově závislé části kódu	37
3	Nově navržená datová struktura pro uchování konfiguračních dat	40
4	Příklad volby nepřekrývajících se číselných reprezentací položek výčtových typů <i>enum</i>	41

1 Úvod

Malé průmyslové žehliče jsou stroje určené k žehlení v hotelích, nemocnicích, veřejných prádelnách a podobných zařízeních.

1.1 Princip funkce žehliče

Princip jejich funkce spočívá v přitlačení žehleného prádla na dutý ocelový válec, který se pohybuje zadanou rychlostí a zevnitř je vytápěn na zadanou teplotu.

Schématické znázornění funkce žehliče je uvedeno na obrázku 16.

1.2 Konstrukční varianty žehliče

Průmyslový žehlič je vyráběn v několika variantách, které pokrývají většinu požadavků zákazníků. Varianty se liší průměrem žehlicího válce, který může být 35 nebo 50 cm, typem řídicího programu (verze OPL nebo COIN), výbavou žehliče z hlediska jeho funkčnosti a šířkou vstupního podávacího stolu. Typy funkční výbavy žehliče:

- I - základní funkční výbava bez skladače, která disponuje pouze jednosměrným výstupním dopravníkem s výstupem prádla směrem k obsluze
- IR - základní funkční výbava bez skladače doplněná obousměrným výstupním dopravníkem s možností volby výstupu prádla směrem k obsluze či směrem opačným
- IF - plná funkční výbava s podélným skladačem prádla, obousměrným reverzačním výstupním dopravníkem s možností volby výstupu prádla směrem k obsluze či směrem opačným

Možné konstrukční varianty žehliče jsou uvedeny v tabulce 1.

Dále se jednotlivé varianty žehliče mohou lišit typem vytápění žehlicího válce:

- plynové vytápění
- parní vytápění
- elektrické vytápění se dvěma nezávislými obvody topných těles

1.3 Řídicí jednotka žehliče

Řídicí jednotka žehliče je zakázkovou aplikací, která byla vyvinuta externí firmou. Její hlavní částí je 16-ti bitový mikroprocesor FUJITSU MB90F407G [1], který disponuje 64 kB paměti ROM pro uložení řídicího programu a 2 kB paměti RAM pro zásobník a uložení proměnných řídicího programu. Aplikace využívá pro běh řídicího programu signály dvou přerušení (jedno pro generování základního signálu časovačů s intervalem 1 ms a druhého pro čítač pulzů z indukčního čidla, který umožňuje měřit délku prádla při

Průměr válce [cm]	Typ řídicího programu	Typ funkční výbavy	Šířky vstupního stolu [cm]
35	OPL	I	140, 160, 200
35	COIN	I	140, 160, 200
35	OPL/COIN	I	140, 160, 200
50	OPL	I	160, 200, 250, 320
50	COIN	I	160, 200, 250, 320
50	OPL/COIN	I	160, 200, 250, 320
50	OPL	IR	200, 250, 320
50	OPL	IF	200, 250, 320

Tabulka 1: Tabulka konstrukčních variant průmyslového žehliče

skládání). Dále jsou pro činnost řídicího programu využity dva generátory pulzního signálu (jeden pro generování zvuku na připojené externí sirénce a druhý pro pulzní modulaci signálu ovládajícího rychlost frekvenčního měniče). Pro ukládání konfiguračních dat je použita EEPROM paměť o velikosti 1024 slov s délkou 16 bitů.

Z hlediska vstupů a výstupů je řídicí jednotka vybavena:

- 14-ti logickými vstupy určenými pro komunikaci s periferiemi, jeden z nich však momentálně není využit, více viz tabulka 2
- čtyřmi logickými vstupy pro čtení stavů jednoho sloupce klávesnice ovládacího panelu stroje
- jedním analogovým vstupem pro měření teploty
- 13-ti logickými výstupy pro ovládání základních funkcí stroje, dva z nich však momentálně nejsou využity, více viz tabulka 3
- pěti logickými výstupy pro určení sloupce klávesnice, jehož stav bude načítán
- jedním analogovým výstupem pro ovládání frekvenčního měniče pohonu stroje pulzní modulací
- portem sériové komunikace pro nahrání řídicího programu, jeho ladění nebo komunikaci s řídicím programem

1.4 Klávesnice řídicí jednotky

Klávesnice řídicí jednotky je základním ovládacím prvkem celého stroje. Její vzhled pro funkční variantu IF je uveden na obrázku 1.

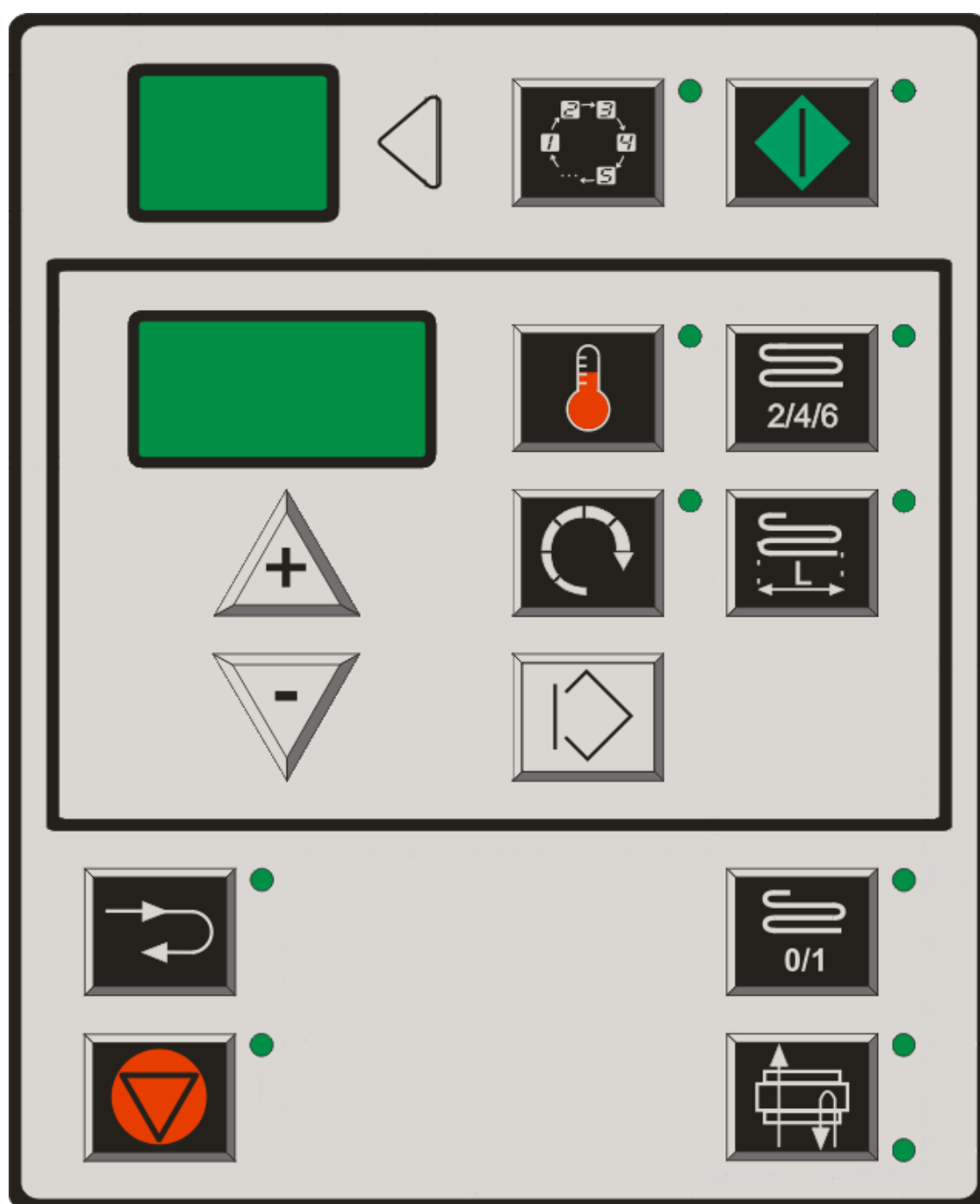
V levé horní části klávesnice jsou umístěny dva displeje pro zobrazování informací, chyb a parametrů žehlení, horní displej zobrazuje dva, spodní tři znaky. Popis a význam jednotlivých kláves je uveden v tabulce 4.

Vstup	Typ stroje	Popis funkce
1	všechny	signál z frekvenčního měniče, že pohon běží
2	všechny	stav kliky ručního pohonu (zasunuta/nezasunuta)
3	všechny	signál použití pedálu nožní spojky vstupních pásů
4	všechny	signál běhu ventilátorů
5	všechny	signál klíče odemčení programovacího módu
6	všechny	signál bezpečnostního podtlakového spínače
7	všechny	signál chyby plynové zapalovací jednotky
8	všechny	nepoužito
9	I	signál mincovníku pro minci s první hodnotou
9	IF	detekce vstupu prádla do skladače
10	IF	signál indukčního čidla pro určení délky prádla
11	I	signál mincovníku pro minci s druhou hodnotou
11	IF	signál předního zažehlovacího válce - je přitisknut k reverzačnímu dopravníku
12	IF	signál zadního zažehlovacího válce - je přitisknut k reverzačnímu dopravníku
13	I	signál klíče pro volbu OPL nebo COIN pro stroj ve variantě OPL/COIN
14	všechny	signál lišty ochrany prstů














Tabulka 2: Seznam a popis logických vstupů řídicí jednotky

Výstup	Typ stroje	Popis funkce
1	všechny	spínání plynového, parního nebo 1. okruhu elektrického topení
2	všechny	spínání 2. okruhu elektrického topení nebo reset zapalovací jednotky plynového topení
3	všechny	spínání frekvenčního měniče pohonu žehliče
4	všechny	ovládání spojky vstupního stolu
5	IF	ovládání pohybu balancéru s zažehlovacími válci
6	všechny	spínání ventilátorů
7	všechny	určení směru pohybu frekvenčního měniče pohonu žehliče
8	všechny	signál povolení činnosti frekvenčního měniče pohonu žehliče
9	I (OPL/COIN), IF	volba druhé sady parametrů frekvenčního měniče
10	IR, IF	ovládání směru pohybu výstupního stolu
11	I (OPL/COIN)	spínač externího zapisovače hodin běhu v OPL módu
12	všechny	nepoužito
13	všechny	nepoužito

Tabulka 3: Seznam a popis logických výstupů řídicí jednotky



Obrázek 1: Vzhled klávesnice žehliče pro funkční variantu IF

Klávesa	Název	Popis funkce
	PROGRAM	Volba programu
	START	Spuštění stroje, zrušení chybového hlášení
	TEPLOTA	Zobrazení a editace teploty
	POČET SKLADŮ	Zobrazení a editace počtu skladů
	PLUS	Zvýšení hodnoty editovaného parametru
	RYCHLOST	Zobrazení a editace rychlosti
	DÉLKA SKLADU	Zobrazení a editace délky skladu
	MINUS	Snížení hodnoty editovaného parametru
	ULOŽENÍ PROGRAMU	Uložení změn provedených v programu, vstup do inicializace po zapnutí stroje
	REVERZACE	Spuštění reverzačního módu chodu stroje
	SKLÁDÁNÍ ZAP/VYP	Povolení nebo zakázání funkce skládání prádla
	STOP	Zastavení stroje, zrušení chybového hlášení
	SMĚR VYVÁŽENÍ PRÁDLA	Změna směru vyvážení prádla na výstupu stroje

Tabulka 4: Seznam a popis tlačítek klávesnice řídící jednotky

1.5 Princip funkce skládání IF varianty žehliče

Princip funkce skládání prádla je znázorněn ve čtyřech fázích na obrázku 17.

1. **fáze** - prádlo je detekováno na vstupu skladače a je pokládáno na výstupní dopravník, který se pohybuje směrem k nakonfigurovanému směru výstupu prádla, balancér se zažehlovacími válci je přitisknut na opačném konci, aby nebránil pokládání prádla
2. **fáze** - měřením délky prádla na vstupu skladače byla zjištěna délka jednoho skladu, směr pohybu výstupního dopravníku je změněn a překlopen je také balancér
3. **fáze** - po opětovném změření délky prádla rovné délce skladu je změněn směr pohybu výstupního zásobníku a překlopen balancér, takže jeho zažehlovací válec vyžehlí hranu na právě provedeném skladu
4. **fáze** - na vstupu skladače již není detekováno prádlo, jeho zbytek je volně položen na poskládanou část a výstupní dopravník ho vyveze nakonfigurovaným směrem

2 Analýza stávajícího zdrojového kódu

Zdrojový kód řídicího programu je psán v jazyce C a prošel již řadou změn. Tou hlavní byla alespoň částečná strukturalizace, neboť původní zdrojový kód dodaný spolu s řídicí deskou nesplňoval ani základní požadavky na strukturalizaci, využíval pouze minimálně funkcí a podprogramů, bylo v něm mnoho skoků na návěští, které přehlednost kódů téměř neumožňovaly. Proto byl kód rozdělen do jednotlivých funkcí a v této podobě udržován až do stávajícího stavu.

Ani tento stav však není uspokojivý, neboť kompletní zdrojový kód řídicího programu je uložen v jednom souboru kódu a jednom hlavičkovém souboru, což příliš neusnadňuje orientaci v celém kódu, a kód řídicího programu stále nesplňuje firemní požadavky pro strukturovanost a přehlednost.

Proto byla provedena analýza stávajícího zdrojového kódu řídicího programu, aby byly odhaleny a napraveny výše popsané nedostatky, zejména pak:

- strukturalizace programu dle obecných zásad programování
- modularizace zdrojového kódu - rozdělení kódu do menších modulů a souborů s ohledem na jejich funkci a určení
- využití stavových automatů pro definici stavů stroje dle specifikace jeho funkcí
- využití datových struktur pro ukládání konfiguračních dat s možností ověření jejich integrity formou kontrolního součtu
- dostatečné komentáře ve zdrojovém kódu s označením data a autora provedené změny

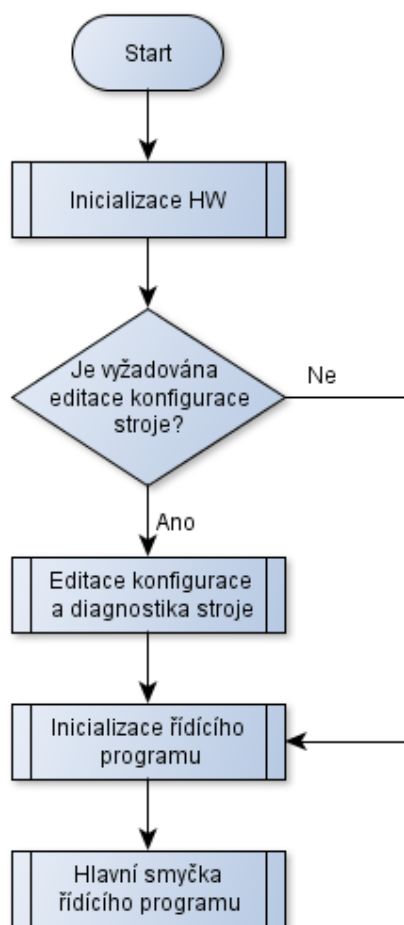
Základní struktura stávajícího zdrojového kódu se sestává ze čtyř částí (viz vývojový diagram na obrázku 2) - inicializace hardware, konfigurace stroje a diagnostické nástroje, inicializace řícího programu a samotná hlavní prováděcí smyčka ovládání a běhu stroje.

Část inicializace hardware provede veškerá nastavení nutná pro správnou funkci vstupů, výstupů a časovačů.

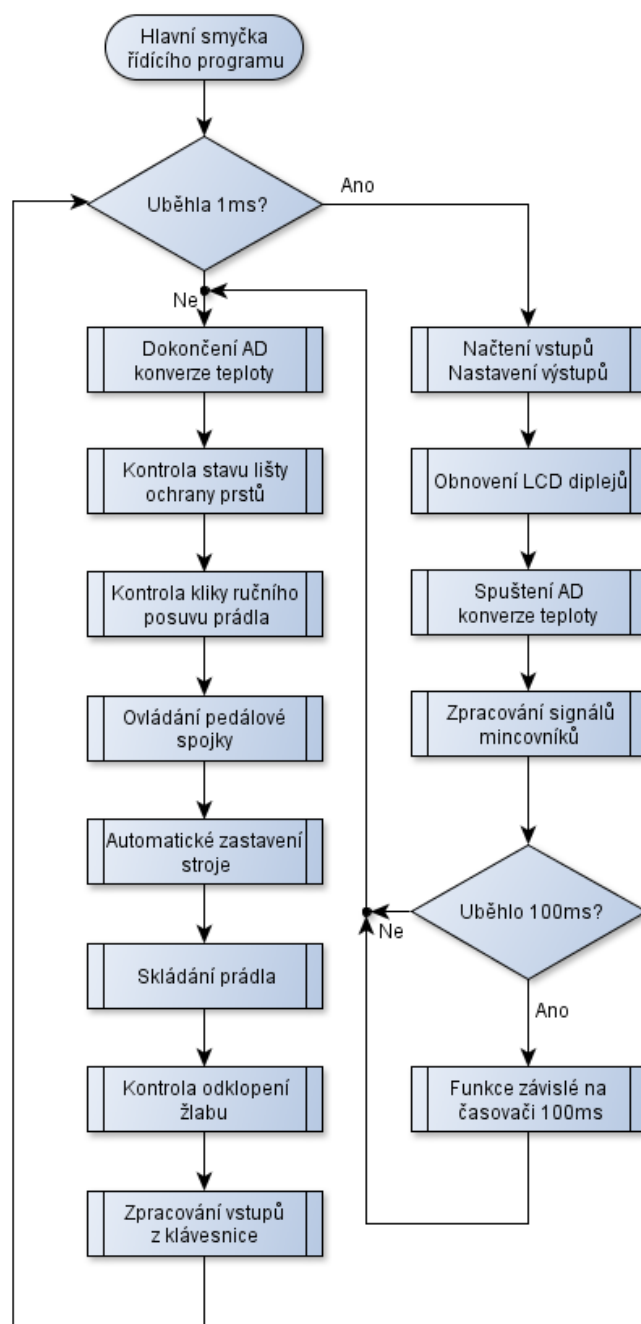
Vstup do konfigurace je povolen pouze při zapnutém klíči programování a stisknutém tlačítku *ULOŽENÍ PROGRAMU*. V konfiguračním menu lze nastavit parametry stroje jako je typ vytápění, funkční výbava stroje, hodnoty mincí mincovníkového stroje apod. (podrobněji viz Programovací manuál [2]). Poslední položkou konfiguračního menu je vstup do diagnostické části, která umožňuje servisním technikům otestovat vstupy a výstupy řídicí jednotky.

V inicializační části řídicího programu jsou načtena veškerá nastavení stroje a parametry defaultního programu označeného číslem nula.

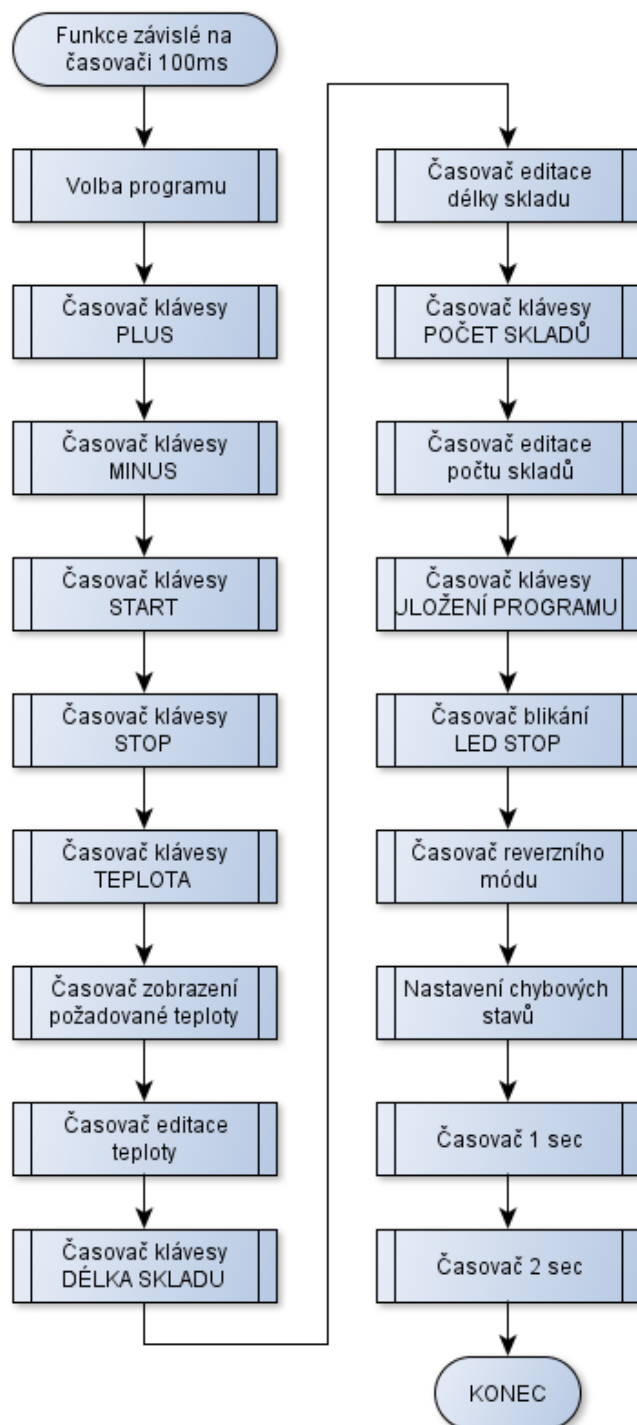
Hlavní prováděcí smyčka řídicího programu (vyobrazena na vývojovém diagramu na obrázku 3) obsahuje několik podprogramů, které jsou prováděny v každém běhu smyčky, další část podprogramů je prováděna na základě signálu přerušení generovaného každou milisekundu a poslední část podprogramů je prováděna každých 100ms (viz vývojový diagram na obrázku 4).



Obrázek 2: Vývojový diagram stávajícího řídicího programu



Obrázek 3: Vývojový diagram hlavní běhové smyčky řídicího programu



Obrázek 4: Vývojový diagram funkcí stávajícího řídicího programu závislých na časovači 100 ms

2.1 Zjištěné nedostatky stávajícího zdrojového kódu řídicího programu

Jak již bylo zmíněno v předchozí kapitole, jedním z nedostatků stávajícího zdrojového kódu je absence jeho modularizace a přítomnost veškerého kódu v jednom zdrojovém souboru.

Téměř veškerá logika ovládání stroje je namísto obsluhy stavových automatů vložena v obslužných funkcích signálů klávesnice. Tato logika je však nepřehledná a obtížně udržovatelná, neboť je tvořena velkým množstvím podmínek a testů na množství logických příznaků nebo dokonce stavů LED klávesnice indikujících typ činnosti stroje.

Stavové automaty jsou v kódu použity pouze při editaci konfigurace stroje a v diagnostickém menu. Konfigurace sama však není tvořena ucelenou strukturou, ale jednotlivé parametry jsou vyčítány z fixních adres z EEPROM a není nijak zajištěna její kompaktnost a validita.

Diagnostické menu neumožňuje přímou kontrolu změnu stavu jednotlivých vstupů, servisní technik musí při diagnostice vstupů přecházet na následující nebo předchozí položku v diagnostickém menu a zpět, aby změnu stavu vstupu ověřil.

Pro velkou část funkcí přímo obsluhujících vstupy z klávesnice, editaci parametrů žehlení je nutno využívat časovače. Jejich použití však není optimální, neboť jak je vidno z vývojového diagramu na obrázku 4, pro kontrolu délky stisku každé klávesy určené pro editaci některého z parametrů je použito zvláštní funkce, která využívá vlastní globální proměnnou pro uložení času stisku. Přitom z principu editace parametrů žehlení vyplývá, že editovat lze vždy pouze jeden parametr. Práce s příznaky předchozích stavů vstupů nebo kláves klávesnice není vždy řešena optimálně. V některých částech kódu je využito datové struktury *PORT* (je uvedena ve výpisu zdrojového kódu 1), která umožňuje jednoduchý přístup k jednotlivým bitům bajtu a pro ukládání příznaků a logických stavů, ale toto pravidlo není dodrženo všude v kódu a tak je možné najít mnoho logických příznaků, které jsou uloženy v datovém typu `unsigned char`, což je vzhledem k omezené kapacitě paměti RAM nevhodné.

Detailní analýzou stávajícího kódu řídicího programu byly také odhaleny a následně opraveny závažné chyby v implementaci. Týkaly se nesprávně deklarovaných polí pro filtraci hodnot teploty žehlicího válce získávaných z analogového vstupu a následného přístupu k hodnotám těchto polí. Tyto chyby byly důvodem dosud neodhaleného a neopraveného přepisování některých globálních proměnných.

```
typedef union {  
    unsigned char byte;  
    struct {  
        unsigned char A : 1;  
        unsigned char B : 1;  
        unsigned char C : 1;  
        unsigned char D : 1;  
        unsigned char E : 1;  
        unsigned char F : 1;  
        unsigned char G : 1;  
        unsigned char H : 1;  
    } bit;  
} PORT;
```

Výpis 1: Datová struktura PORT pro ukládání logických stavů a příznaků

3 Přepřacování řídící logiky ovládání stroje

Řídící logika stroje byla kvůli vyšší přehlednosti a v souladu s firemními požadavky pro strukturovanost a přehlednost přepřacována do formy stavových automatů.

Obsluha hlavní smyčky řízení stroje byla z hlavní funkce `main` (viz vývojový diagram na obrázku 5) vyjmuta a zařazena jako samostatná funkce. Ta je z hlavní funkce v nekonečném cyklu volána. Toto oddělení bylo nutné pro provázání samotného řídicího software s GUI nádstavbou (viz kapitola 6), neboť tatáž funkce je cyklicky volána z vlákna vytvořeného pro běh řídicího software pod GUI nádstavbou. Vývojový diagram hlavní smyčky řízení stroje je uveden na obrázcích 6 a 7.

3.1 Obsluha hlavní smyčky řízení stroje

Na počátku hlavní řídící smyčky jsou zaktualizovány stavy signálů veškerých časovačů a dle nich pak provedeny nutné funkce pro obsluhu vstupů, výstupů, displejů a stavových LED. Následně je obsloužen hlavní stavový automat (viz kapitola 3.2), dokončena AD konverze hodnoty teploty, obsloužen stavový automat zvuku (viz kapitola 3.8), a nastaveny příznaky předchozích stavů kláves a vstupů.

3.2 Obsluha hlavního stavového automatu

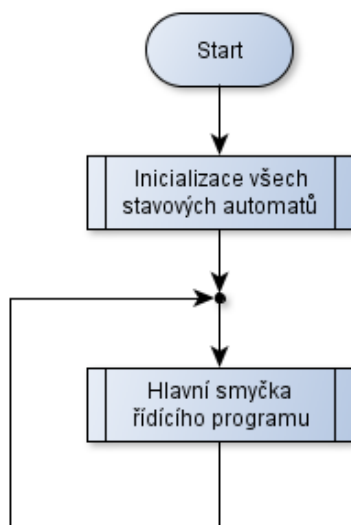
Hlavní stavový automat řídicího programu rozlišuje základní stavy stroje a je vrcholem celé hierarchie stavových automatů celkové logiky řízení stroje. Diagram hlavního stavového automatu je uveden na obrázku 8 a popis jednotlivých stavů v tabulce 5.

Č. stavu	Popis
1	Obsluha stavového automatu inicializace řídicího programu (viz kapitola 3.3)
2	Nastavení parametrů defaultního programu č. 0
3	Obsluha stavového automatu běhu stroje (viz kapitola 3.6)
4	Obsluha zpětného chodu stroje vyvolaného stiskem lišty ochrany prstů
5	Obsluha chybových stavů
6	Zpětný chod stroje vyvolaný uživatelem z klávesnice (pouze ve verzi OPL)
7	Obsluha chodu stroje při zvednutém vstupním žlabu (pro stroje s funkční výbavou IR nebo IF)

Tabulka 5: Popis stavů hlavního stavového automatu

3.3 Obsluha stavového automatu inicializace řídicího programu

Stavový automat inicializace řídicího programu provádí veškeré nutné kroky, které je potřeba vykonat před započítím normálního běhu stroje. Diagram tohoto stavového automatu je uveden na obrázku 9, tabulka 6 uvádí popis jednotlivých jeho stavů.



Obrázek 5: Vývojový diagram hlavní funkce *main* přepracovaného řídicího programu

Č. stavu	Popis
1	Inicializace hardware
2	Načtení a kontrola správnosti konfiguračních dat řídicího programu
3	Kontrola požadavku na editaci konfiguračních dat
4	Editace konfiguračních dat vyžádaná uživatelem nebo nutná z důvodu jejich nesprávnosti
5	Obsluha stavového automatu editace konfiguračních dat (viz kapitola 3.4)
6	Zobrazení verze řídicího programu
7	Zobrazení nakonfigurovaného typu vytápění a šířky vstupních pásů
8	Kontrola funkčnosti lišty ochrany prstů
9	Nefunkční lišta ochrany prstů

Tabulka 6: Popis stavů stavového automatu inicializace řídicího programu

3.4 Obsluha stavového automatu editace konfiguračních dat

Navržený stavový automat editace konfiguračních dat zehliče zohledňuje veškeré volby konfigurace stroje podobně, jako tomu je u stávající verze řídicího programu. Doplněny byly některé nové položky, které vzešly z doplnění a rozšíření řídicího software na základě požadavků zákazníků a připomínek servisních techniků. Tyto nové volby jsou podrobněji popsány v kapitole 5. Obsluha stavového automatu editace konfiguračních dat byla navíc doplněna o automatická nastavení voleb, které v závislosti na provedené změně konfigurace nejsou uživateli přístupná, ale jejich správná hodnota je nutná pro celkovou validitu konfiguračních dat.

Jak je zřejmé z diagramu na obrázku 10, přechod mezi jednotlivými položkami je prováděn po stisku klávesy START a to pouze v jednom směru, aby byla zajištěna platnost konfiguračních dat. Pro celkovou přehlednost diagramu nejsou signály pro změnu hodnoty konfigurační položky zobrazeny. Tato změna hodnoty je provedena jako reakce na stisk klávesy PLUS nebo MINUS u všech stavů, kromě informací o době běhu stroje a době topení, jak je uvedeno v tabulce 7.

Č. stavu	Popis
1	Volba druhu vytápění stroje
2	Volba funkčního vybavení stroje
3	Volba šířky vstupních podávacích pásů
4	Hodnota teplotní hystereze (viz kapitola 5.2)
5	Volba vybavení stroje pedálem spojky
6	Povolení volby verze OPL nebo COIN za běhu řídicího programu (viz kapitola 5.3)
7	Konfigurace verze OPL nebo COIN
8	Konfigurace délky času žehlení pro I. typ mince mincovníku
9	Konfigurace délky času žehlení pro II. typ mince mincovníku
10	Konfigurace minimálního zaplaceného času žehlení pro start vytápění stroje
11	Konfigurace teploty žehlení stroje verze COIN
12	Konfigurace rychlosti žehlení stroje verze COIN
13	Konfigurace vzdálenosti reverzního dopravníku od vstupu skladače; ovlivňuje přesnost prvního skladu prádla (více viz manuál žehliče [2])
14	Konfigurace opravy počtu pulzů indukčního čidla; ovlivňuje přesnost měření délky prádla a tím kvalitu a preciznost skládání (více viz manuál žehliče [2])
15	Konfigurace redukce délky skladu; ovlivňuje kvalitu skládání prádla (více viz manuál žehliče [2])
16	Konfigurace akcelerace reverzního dopravníku při skladu prádla; ovlivňuje kvalitu skládání prádla (více viz manuál žehliče [2])
17	Volba povolení či zakázání možnosti volby směru vyvážení prádla
18	konfigurace speciálního továrního kódu
19	Zobrazení doby běhu stroje ve stovkách hodin, tuto hodnotu nelze editovat
20	Zobrazení doby běhu topení stroje ve stovkách hodin, tuto hodnotu nelze editovat
21	Obsluha stavového automatu diagnostiky stroje (viz kapitola 3.5)

Tabulka 7: Popis stavů stavového automatu editace konfiguračních dat

3.5 Obsluha stavového automatu diagnostiky stroje

Diagnostika stroje umožňuje zkontrolovat funkci logických i analogových vstupů a výstupů řídicí jednotky. Diagram uvedený na obrázku 11 byl vzhledem k snadno pochopitelné funkcionalitě zjednodušen a zkrácen na čtyři stavy, jak je uvedeno v tabulce 8. Stiskem klávesy PLUS je proveden přechod na následující položku diagnostiky, stiskem klávesy MINUS na předchozí. Testování funkcionality výstupů je ovládáno klávesou START, při jejímž stisku je daný výstup sepnut, jejím uvolněním rozepnut. Při testování funkčnosti analogového výstupu jsou kromě nastavení hodnoty tohoto výstupu sepnuty i potřebné logické výstupy, které ovládají chod frekvenčního měniče, aby bylo zajištěno spuštění pohonu a tím mohla být ověřena jeho funkčnost. Vzhledem k oddělení zpracování hodnot vstupů řídicí jednotky od samotné logiky řízení stroje, jeho konfigurace i diagnostiky, bylo docíleno automatické aktualizace hodnot vstupů při jejich diagnostice a odstraněna nutnost přechodu na další či předchozí položku diagnostiky pro aktualizaci stavu či hodnoty vstupu ve stávající verzi řídicího programu.

Č. stavu	Popis
1	Diagnostika logických vstupů (1 - 14, vstup 10 nelze přímo testovat, je na něj připojeno indukční čidlo)
2	Diagnostika analogového vstupu použitého pro měření teploty žehlicího válce
3	Diagnostiky logických výstupů (1 - 13)
4	Diagnostika analogového výstupu pro ovládání frekvenčního měniče pohonu stroje

Tabulka 8: Popis stavů stavového automatu diagnostiky stroje

3.6 Obsluha stavového automatu běhu stroje

Stavový automat běhu stroje obsluhuje základní stavy žehliče v průběhu normální práce na stroji. Před samotnou obsluhou tohoto stavového automatu je provedena obsluha stavového automatu mincovníku (viz kapitola 3.7), neboť dva stavy automatu běhu stroje jsou určeny výlučně pro případ, že je stroj nakonfigurován do verze COIN a tyto dva stavové automaty spolu kooperují. Tato kooperace je nutná pro zajištění maximální efektivity činnosti stroje i finančního zisku pro majitele stroje ve veřejné prádelně.

Je-li stroj nakonfigurován do verze OPL, jsou namísto obsluhy stavového automatu mincovníku zpracovávány požadavky na úpravu parametrů žehlení.

Důležitým stavem normálního běhu stroje je stav označený číslem 3 (viz obrázek 12 a tabulka 9), který zajišťuje automatické bezpečné zastavení stroje po ochlazení žehlicího válce na teplotu, při níž nedojde k poškození žehlicích pásů.

Č. stavu	Popis
1	Stroj je zastaven a připraven k započetí vytápění a žehlení nebo vychlazen k bezpečnému vypnutí
2	Stroj běží na konfigurovanou rychlost včetně vstupních pásů a vytápí na konfigurovanou teplotu
3	Stroj se chladí na teplotu bezpečnou pro vypnutí a zastavení stroje, topení neběží, pohon stroje běží na nejnižší rychlost
4	Pouze ve verzi COIN: Stroj se vytápí na konfigurovanou teplotu žehlení, vstupní pásy jsou zastaveny, pohon běží na nejnižší rychlost
5	Pouze ve verzi COIN: Stroj je vytopen na požadovanou rychlost, odpočet času pro žehlení byl započat, pohon běží stále na nejnižší rychlost a vstupní pásy stojí

Tabulka 9: Popis stavů stavového automatu běhu stroje

3.7 Obsluha stavového automatu mincovníku

Obsluha stavového automatu mincovníku zajišťuje generování signálů potřebných pro přechod mezi stavy automatu normálního běhu stroje určenými pro COIN verzi ovládání žehliče. Tento jednoduchý stavový automat je uveden na obrázku 13 a popis jeho stavů v tabulce 10.

Č. stavu	Popis
1	Nebyl vložen dostatek mincí pro započetí žehlení
2	Vložena alespoň minimální hodnota mincí pro započetí vytápění stroje a žehlení
3	Stroj byl vytopen na požadovanou teplotu a odpočet zaplaceného času běží

Tabulka 10: Popis stavů stavového automatu mincovníku

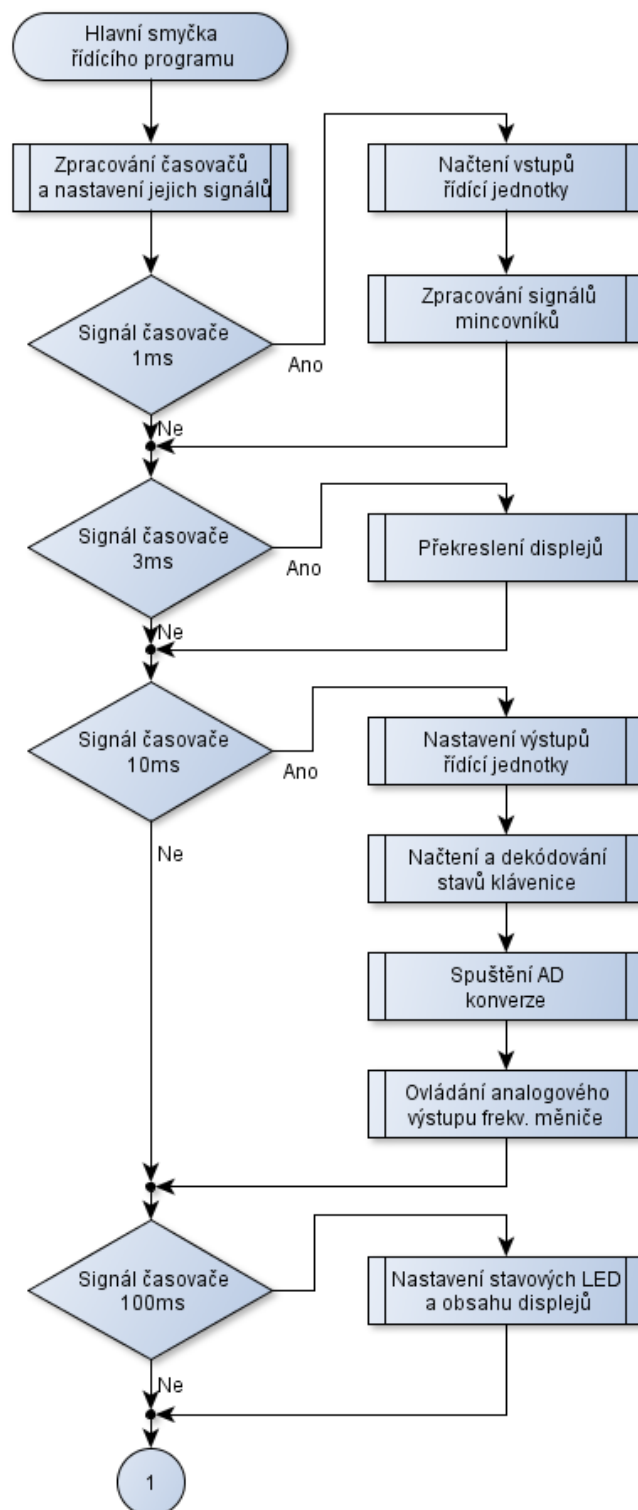
3.8 Obsluha stavového automatu zvuku

Generování varovných zvukových signálů je obsluhováno samostatným stavovým automatem, který v závislosti na požadavcích řídicí logiky stroje zajišťuje:

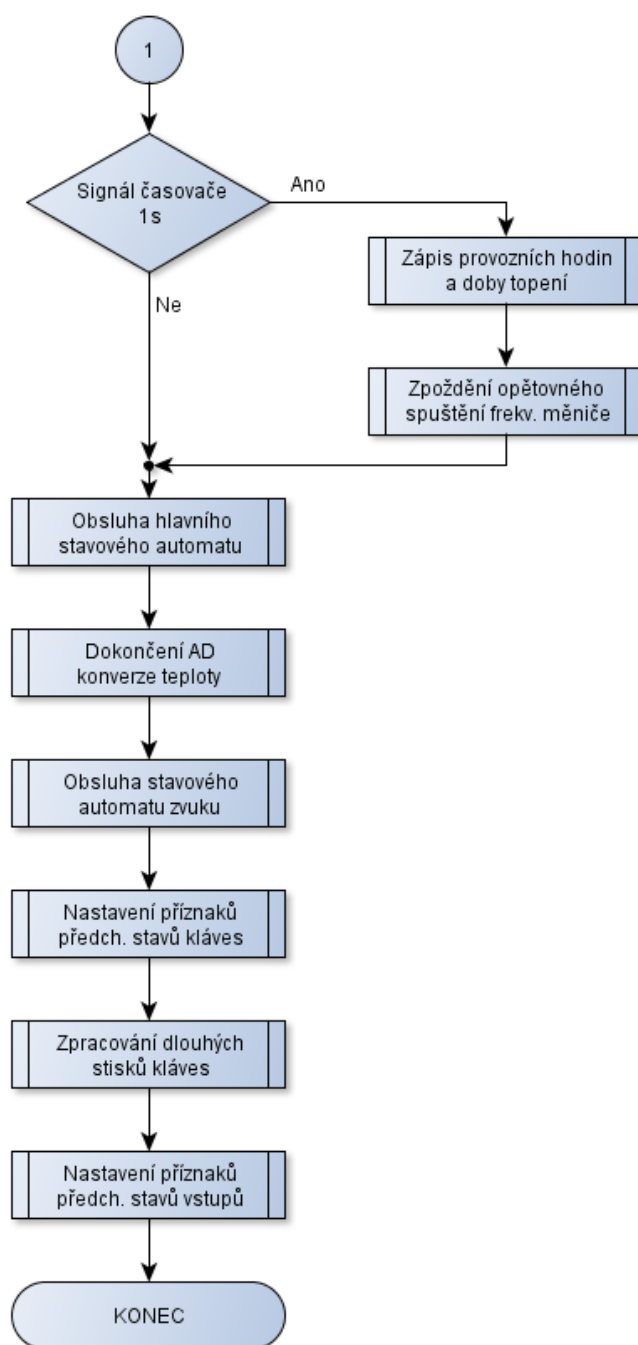
- *varovný signál o délce 1 vteřiny* pro upozornění obsluhy COIN verze stroje:
 - na vložení dostatečné hodnoty mincí pro započetí žehlení
 - na započetí odpočítávání poslední minuty zaplaceného času žehlení
- *opakovaný varovný signál o délce 1 vteřiny následovaný stejnou dobou odmlky* pro notifikaci:

- závažného chybového stavu stroje - signál je ukončen zrušením této chyby stiskem klávesy START nebo STOP
- zpětného chodu stroje vyvolaného stiskem lišty ochrany prstů - signál je ukončen zároveň s dokončením zpětného chodu stroje
- *varovný signál o délce 3 vteřin následovaný opakovanou odmlkou o délce 5 vteřin a signálem o délce 2 vteřin pro upozornění obsluhy COIN verze stroje, že byl vytopen na požadovanou teplotu a započal odpočet zaplaceného času žehlení*

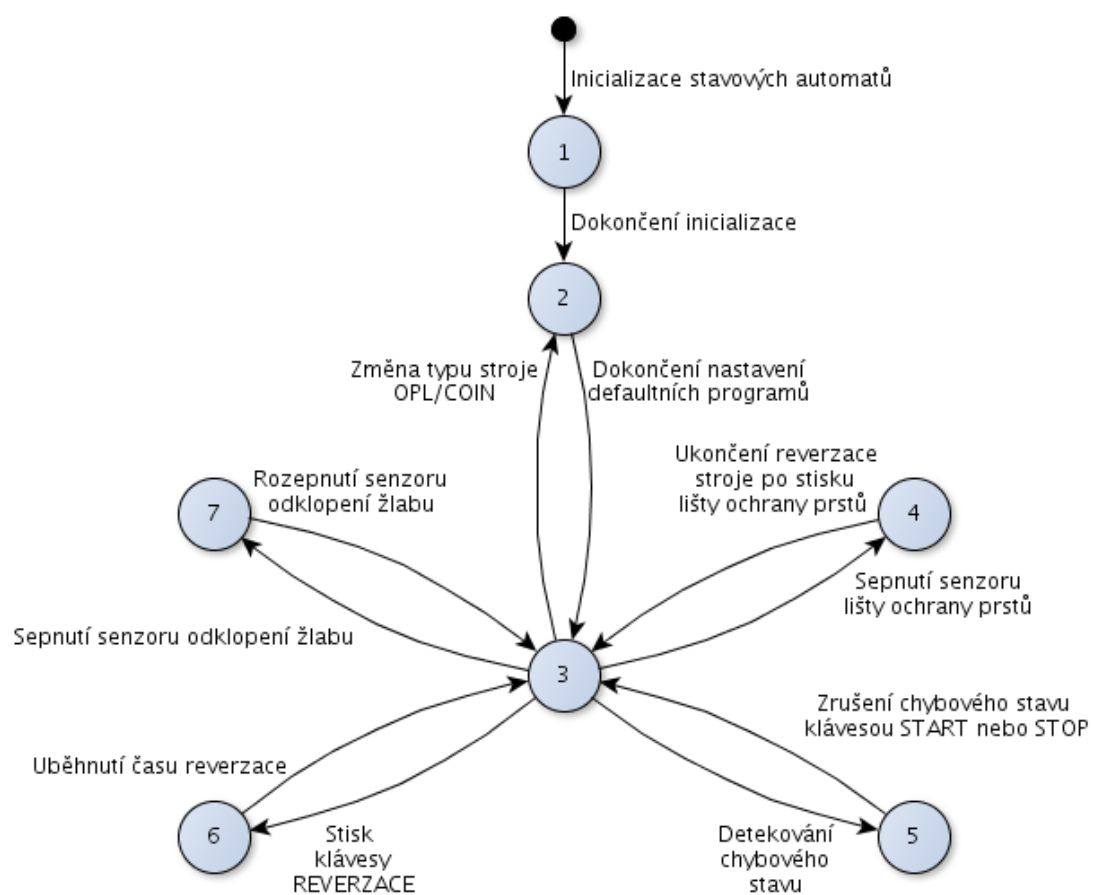
Pro generování jednotlivých zvukových signálů na stávající hardwarové platformě je využito konfigurovatelného generátoru pulzů, který je součástí mikroprocesoru, a umožňuje volit různé frekvence zvukových signálů pro různé typy varování.



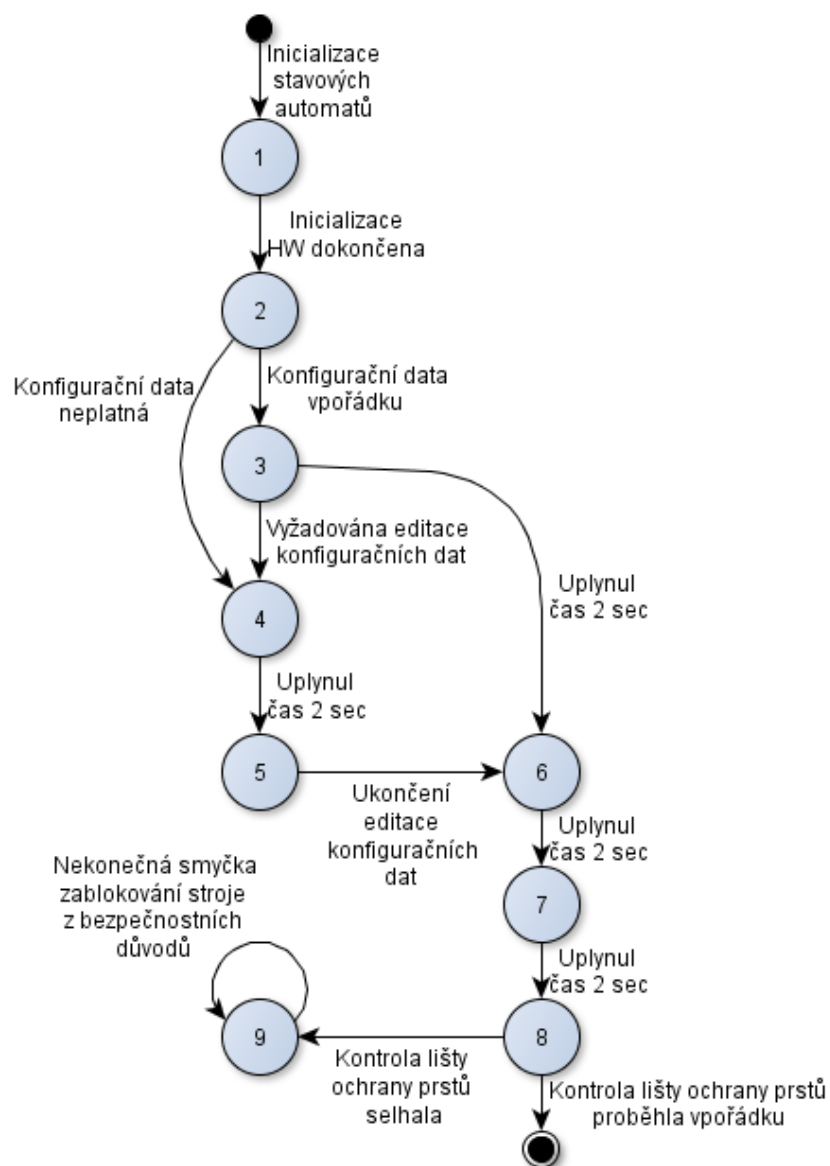
Obrázek 6: Vývojový diagram hlavní prováděcí funkce přepracovaného řídicího programu - část 1



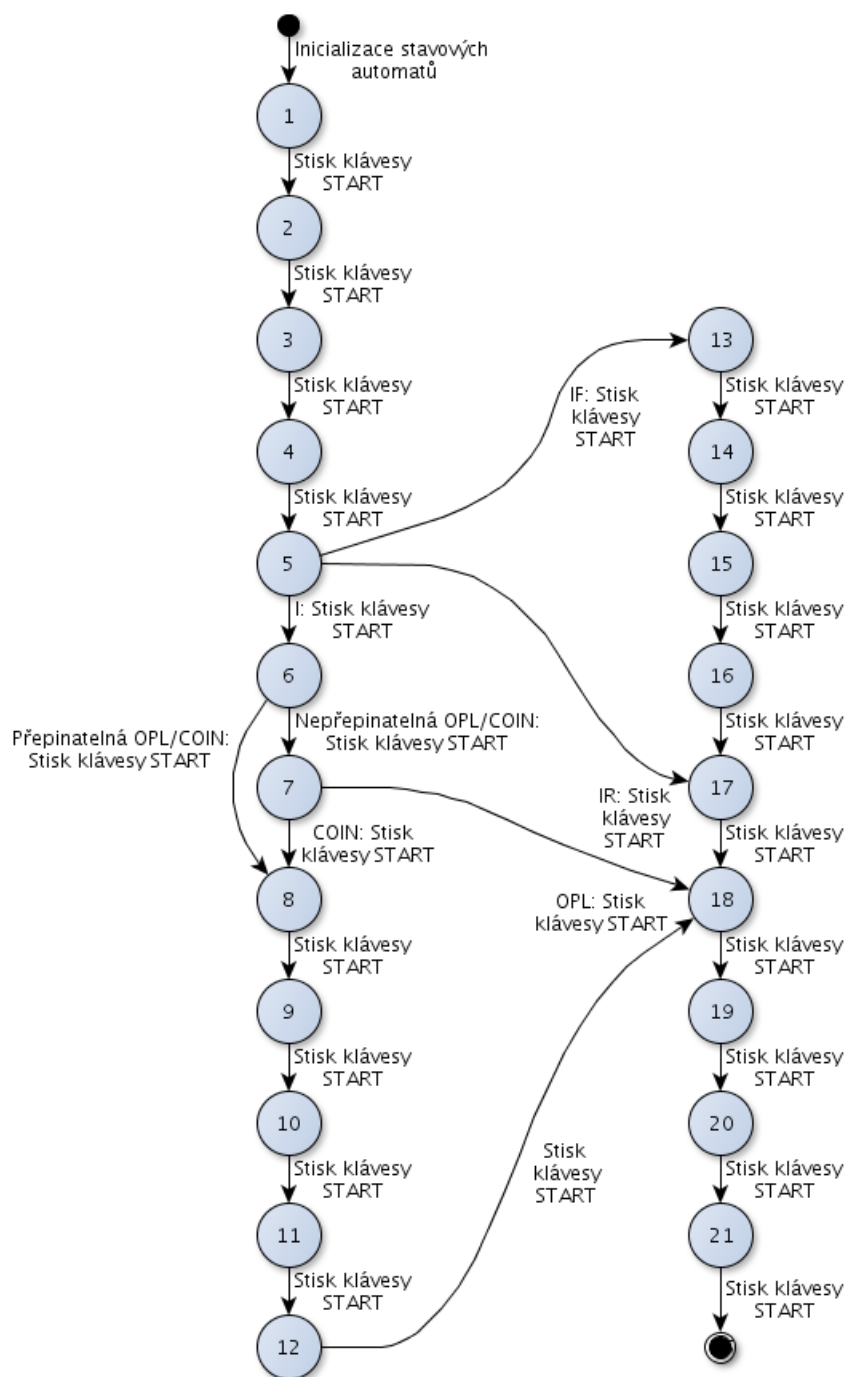
Obrázek 7: Vývojový diagram hlavní prováděcí funkce přepracovaného řídicího programu - část 2



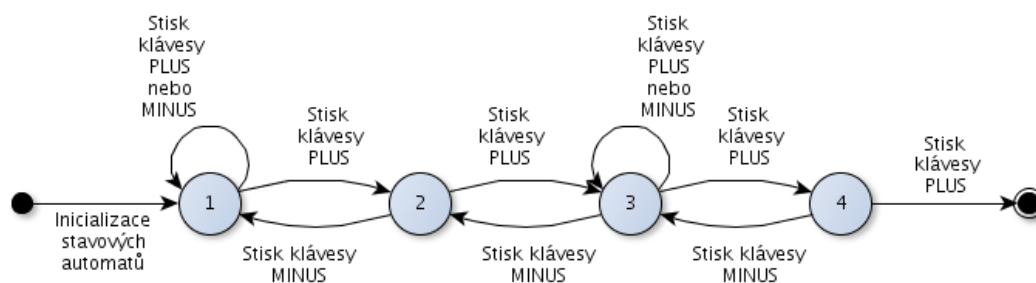
Obrázek 8: Hlavní stavový automat přepracovaného řídicího programu



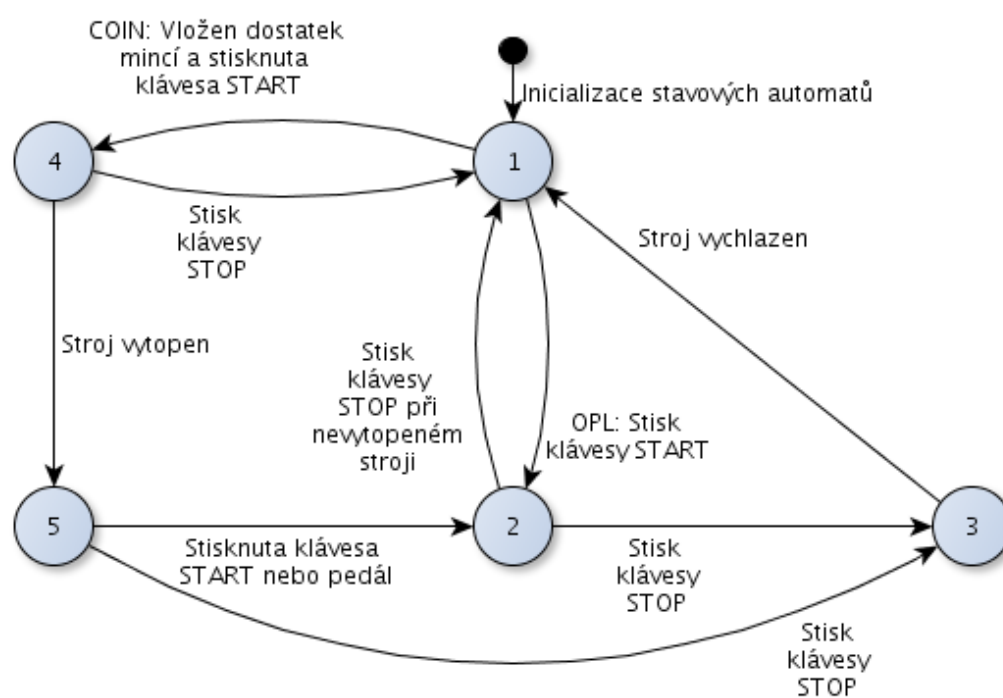
Obrázek 9: Stavový automat inicializace řídicího programu



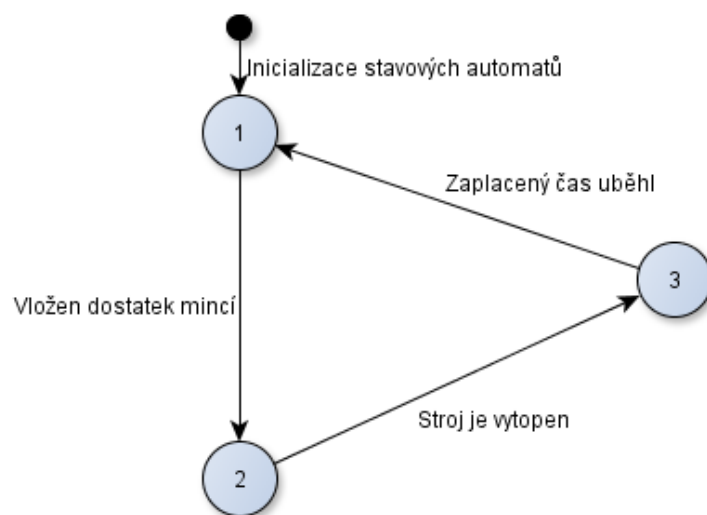
Obrázek 10: Stavový automat editace konfiguračních dat



Obrázek 11: Stavový automat diagnostiky stroje



Obrázek 12: Stavový automat běhu stroje



Obrázek 13: Stavový automat mincovníku

4 Modularizace řídicího programu s ohledem na hardwarové závislosti software

Veškerý zdrojový kód funkcí stávajícího řídicího programu byl rozdělen na jednotlivé moduly podle účelu jednotlivých funkcí. Pro zvýšení přehlednosti byly jednotlivé moduly roztrženy do následujících složek:

config - obsahuje pouze jeden modul v souborech *mach_cfg.h* a *mach_cfg.c*, který obsahuje konfigurační data a funkce pro jejich validaci, nastavení defaultních hodnot a další práci s nimi

func - moduly zajišťující hlavní funkcionality stroje (viz kapitola 4.1)

gui - zdrojové kódy implementace GUI (viz kapitola 6)

hw - moduly a data pro přímou interakci s hardware (viz kapitola 4.2)

V kořenovém adresáři projektu byly ponechány pouze následující zdrojové a hlavičkové soubory:

bool.h - modul s definicí typu `bool` pro jazyk C

errors.h, errors.c - modul pro testování a zpracování chybových stavů stroje

extern.h - hlavičkový soubor s deklaracemi globálních proměnných řídicího programu

init.h, init.c - modul obsahující obsluhy stavových automatů pro inicializaci, konfiguraci a diagnostiku stroje

main.h, main.c - modul s hlavní funkcí řídicího programu a obsluhou hlavních stavových automatů

START.asm, Mb90495.asm - moduly v assembleru nutné pro kompilaci řídicího programu pro stávající platformu

states.h - modul obsahující deklarace všech stavových automatů ve formě výčtových typů `enum`

Dále samozřejmě kořenový adresář obsahuje soubory s definicí projektu pro použité IDE (viz kapitola 6.2).

4.1 Moduly složky *func*

coin.h, coin.c - modul pro obsluhu stavového automatu vyhodnocení plateb mincemi a odpočet času. Odpočet času je vázán pouze na globální časovač mincovníku po 1 vteřině, odpočet minut si podle něj řídí stavový automat sám. Globální časovač (např. jedné minuty) zde nelze použít neboť odpočet času pro žehlení na COIN verzi žehliče neběží například za reverzačního běhu stroje vyvolaného stiskem lišty ochrany prstů

folding.h, folding.c - modul s funkcemi a podprogramy pro podélné skládání prádla na žehličích s funkční výbavou IF

heating.h, heating.c - modul obsahující funkce pro regulaci vytápění žehličích válce pro všechny typy vytápění (elektrické, parní a plynové); pro elektrické vytápění, které je dvouokruhové, řeší zpoždění spínání druhého okruhu po prvním, aby byly minimalizovány proudové špičky vyvolané sepnutím obou okruhů ve stejný časový okamžik

program.h, program.c - modul s funkcemi pro načtení žehličích programů z EEPROM, uložení změn v programu do EEPROM, validaci a změny hodnot parametrů programu žehlení či nastavení jejich defaultních hodnot, obsluhu stavového automatu výběru a editace parametru, obsluhu stavového automatu postupného zobrazení všech platných parametrů nově zvoleného programu

speed.h, speed.c - moduly obsahující funkce pro spouštění nebo zastavení pohonu stroje, regulaci jeho rychlosti a zpoždění spouštění pohonu okamžitě po jeho zastavení - jedná se o bezpečnostní funkcionalitu, která má zabránit poškození frekvenčního měniče pohonu stroje

4.2 Moduly složky *hw*

display.h, display.c - modul obsahující funkce pro zobrazení všech potřebných informací na obou displejích a nastavení LED indikátorů v závislosti na aktuálním stavu stroje, jak je definován v hlavních stavových automatech

eeprom.h, eeprom.c - modul s funkcemi pro zápis a čtení 16-bitového slova nebo bloku dat do/z EEPROM, obsahuje také důležitou funkci, která podle velikosti konfiguračních dat a velikosti dat parametrů programů žehlení určuje adresy jejich uložení v EEPROM, aby nebyla překryta kalibrační data AD převodníku teploty zapisovaná do EEPROM na fixní adresu výrobcem řídicí desky

inputs.h, inputs.c - modul obsahující funkce pro načítání hodnot logických hardwarových vstupů, nastavování jejich příznaků předešlých stavů pro detekci změny hodnoty, provádění AD konverze hodnoty analogového vstupu na teplotu žehličích válce a filtrace těchto hodnot. Modul obsahuje deklaraci symbolických názvů jednotlivých logických hardwarových vstupů a příznaků uložených v datové struktuře *PORT*

interrupts.h, interrupts.c - modul s inicializací a obsluhou přerušení pro stávající hardwarovou platformu

ioports.h, ioports.c - modul obsahující inicializaci a konfiguraci vstupních a výstupních portů a deklaraci datové struktury *PORT* využívané pro ukládání logických hodnot vstupů, výstupů a jejich příznaků

keyboard.h, keyboard.c - modul s funkcemi pro čtení stavů signálů z klávesnice, jejich filtrování a převod na logické signály stisku kláves a příznaků předchozích stavů uložených v datové struktuře *PORT*

outputs.h, outputs.c - modul obsahující funkce pro nastavení hodnot logických výstupů řídicí jednotky a nastavení hodnoty analogového výstupu pro frekvenční měnič pohonu stroje

pulsegen.h, pulsegen.c - modul obsahující konfigurační metody generátorů pulzů na stávající hardwarové platformě

sound.h, sound.c - modul s obsluhou stavového automatu generování zvukových upozornění na dostatečný počet mincí pro započetí žehlení, vytopení žehlícího válce na požadovanou teplotu a poslední minutu žehlení ve verzi COIN, reverzací stroje v důsledku stisku lišty ochrany prstů nebo varování při závažných chybových zprávách pro všechny verze stroje

timers.h, timers.c - modul obsahující funkce pro generování příznaků uplynutí požadovaného času všech časovačů, veškeré časovače jsou odvozeny od základního časovače s intervalem 1 ms generovaného pomocí přerušení

Mb90495.h - module s deklaracemi symbolů pro přístup k jednotlivým vstupně výstupním registrům mikroprocesoru řídicí jednotky, nutné pro kompilaci hardwarově závislých částí na stávající platformě

vectors.c - modul obsahující deklarace prototypů funkcí přerušení, nutné pro kompilaci na stávající platformě

Většina funkcí z výše uvedených modulů komunikuje přímo s hardwarem stávající platformy. Aby byla zajištěna nezávislost těchto funkcí a modulů na hardwarové platformě, jsou veškeré takové části kódu, případně některé celé funkce, ohrazeny příkazy podmíněné kompilace. Podmínkou takové kompilace je vždy identifikátor, který je implicitně nebo explicitně definován pro zvolenou cílovou platformu (viz výpis 2).

```
void StartADConversion() {
    ADConversionStarted = true;
    #if defined(__FUJITSU__)
        ADCS1_STRT = 1; // start prevodu
    #endif // defined(__FUJITSU__)
}
```

Výpis 2: Příklad oddělení hardwarově závislé části kódu

5 Rozšíření a doplnění stávajících funkcí řídicího programu

5.1 Návrh a implementace datové struktury pro uchování konfiguračních dat

Pro uchování konfiguračních dat byla navržena v souladu s firemními požadavky nová datová struktura (viz výpis 3) obsahující všechny položky konfigurace stávajícího řídicího programu a doplněny byly nové položky pro povolení volby verze OPL nebo COIN za běhu stroje (viz kapitola 5.3), teplotní hystereze (viz kapitola 5.2) a kontrolního součtu konfiguračních dat. Tento kontrolní součet je aktualizován před každým uložením konfiguračních dat do EEPROM a při jejich načítání je jeho uložená hodnota porovnávána s kontrolním součtem načtených konfiguračních dat. Tímto způsobem je zaručena kompaktnost konfiguračních dat, jejich validita je pak ještě prověřena kontrolou správnosti hodnot jednotlivých položek konfigurace. V případě, že nejsou konfigurační data kompaktní nebo jejich hodnoty validní, je v řídicím programu automaticky proveden vstup do jejich editace, aby mohla být data korektně nastavena. Pro zajištění maximální účinnosti kontrolního součtu jsou pro jednotlivé hodnoty výčtových typů *enum* zvoleny nepřekrývající se číselné reprezentace jejich hodnot (viz příklad ve výpisu 4).

```

unsigned int crc;                                // kontrolni soucet konfiguracnich dat

enum HEATING_TYPE heating_type;                  // typ topeni
enum MACHINE_EXECUTION machine_execution;         // verze OPL nebo COIN

enum MACHINE_SIZE machine_size;                  // sirka stolu se vstupnimi podavacimi pasy
unsigned char temperature_hysteresis;           // hodnota teplotni hystereze
enum PEDAL_AVAILABLE pedal_available;           // vybava stroje pedalem spojky
enum OPL_COIN_SWITCHABLE opl_coin_switchable;    // povoleni/zakazani prepinari OPL/COIN
                                                    // za behu stroje

enum MACHINE_TYPES machine_type;                 // typ funkcní vybavy zehlice
unsigned char coin1_time;                        // cas zehleni pro minci I.
unsigned char coin2_time;                        // cas zehleni pro minci II.
unsigned char coin_time_start;                   // minimalni zaplacený cas pro start stroje
                                                    // verze COIN

unsigned char coin_temperature;                  // konfigurace teploty zehleni pro verzi COIN
unsigned char coin_speed;                        // konfigurace rychlosti zehleni pro verzi COIN
unsigned char table_distance;                    // vzdalenost reverzniho dopravniku od vstupu
                                                    // skladace, pouze IF

unsigned char pulse_length;                      // korekce pulzu pro urceni delky pradla,
                                                    // pouze IF

unsigned char fold_acceleration;                 // akcelerace reverzniho dopravniku pri skladani
                                                    // pradla, pouze IF

unsigned char fold_reduction;                    // redukce delky skladu pradla, pouze IF
enum OUTPUT_DIRECTION_CHANGE_ENABLE output_direction_enable;
                                                    // povoleni/zakazani zmeny smeru vystupu
                                                    // pradla,
                                                    // pouze IR nebo IF

unsigned int life_hours;                         // doba behu stroje
unsigned int heating_hours;                      // doba topeni stroje
unsigned char factory_code;                     // tovarni kod
} CONFIG;

```

Výpis 3: Nově navržená datová struktura pro uchování konfiguračních dat

```

typedef enum HEATING_TYPE {
    HEAT_GAS = 41,
    HEAT_STEAM = 42,
    HEAT_ELECTRIC = 43
}heating_type_enum;

typedef enum MACHINE_SIZE {
    SIZE_I35_140 = 53,
    SIZE_I35_160 = 54,
    SIZE_I35_200 = 55,

    SIZE_I50_160 = 56,
    SIZE_I50_200 = 57,
    SIZE_I50_250 = 58,
    SIZE_I50_320 = 59,

    SIZE_IR_IF_200 = 60,
    SIZE_IR_IF_250 = 61,
    SIZE_IR_IF_320 = 62
}machine_size_enum;

```

Výpis 4: Příklad volby nepřekrývajících se číselných reprezentací položek výčtových typů *enum*

5.2 Teplotní hystereze

Jednou z nepříjemných vlastností žehliče je zpoždění a setrvačnost ohřevu žehlicího válce. Ta může způsobovat dočasné zvyšování teploty žehlicího válce i po vypnutí vytápění a překročení bezpečné teploty žehlicího válce při zastaveném pohonu stroje. Následkem toho by pak mohlo být spálení a poškození žehlicích pásů. V řídicím programu je na tento fakt pamatováno a je v takovém případě hlášena chyba doprovázená zvukovým signálem.

Aby byl vliv setrvačnosti ohřevu potlačen i během ochlazování žehliče před zastavením, je v řídicím programu zavedena konstanta teplotní hystereze, která zastaví pohon stroje až při poklesnutí teploty žehlicího válce na teplotu o 4°C nižší, než je bezpečná teplota pro zastavení pohonu stroje. Ani tato hodnota však u některých strojů (obzvláště u žehliček s elektrickým vytápěním) není dostačující a způsobuje v některých případech zbytečné výskyty výše popsané chybové zprávy. Naproti tomu přílišné navýšení teplotní hystereze navyšuje náklady na běh stroje ve stavu ochlazování.

Vzhledem k tomu, že hodnota potřebné teplotní hystereze je závislá na okolním prostředí stroje, byla tato přidána jako položka konfigurace a uživatel stroje si ji v rozmezí 4 - 10°C může nastavit dle svých potřeb.

5.3 Volba verze OPL nebo COIN za běhu stroje

Pro jednoho z významných zákazníků byla na jeho požadavek vyvinuta speciální verze stávajícího řídicího programu pro funkční výbavu I, která umožňuje přechod mezi ver-

zemi obsluhy stroje OPL a COIN za chodu stroje bez nutnosti jeho restartu a změny konfiguračních dat. Tato změna je prováděna změnou stavu logického vstupu č. 13 (viz tabulka 2) a je realizována klíčovým spínačem. Změna stavu tohoto logického vstupu je zpracovávána pouze v případě, že je stroj zastaven nebo běží automatické ochlazování žehlicího válce před zastavením stroje (viz tabulka 9).

Existence dvou velmi podobných verzí řídicího programu je však neefektivní jak z pohledu údržby jejich zdrojových kódů, tak z pohledu produkce těchto strojů a proto bylo rozhodnuto sloučit je do jednoho zdrojového kódu. Aby byla zajištěna korektní funkčnost obou verzí řídicího programu, byla konfigurační data doplněna o položku, která povoluje nebo zakazuje funkcionalitu přepínatelné verze OPL/COIN (viz tabulka 7).

6 GUI nádstavba řídicího programu

Jelikož v průběhu práce na přepracování řídicího programu nebylo dokončeno výběrové řízení nového RT systému, bylo proto rozhodnuto vypracovat pro řídicí program nádstavbu s GUI, která bude pracovat na platformách Microsoft Windows a Linux. Tato nádstavba má umožnit simulaci funkcí stroje s maximální možnou shodou chování řídicího programu na stávajícím reálném HW. Velkou výhodou tohoto řešení je úspora času jak v průběhu přepracování řídicího programu žehliče, tak při jeho budoucích úpravách a vylepšeních.

Platforma Microsoft Windows byla zvolena z důvodu vybavenosti firemních počítačů tímto systémem, platforma Linux pro ověření funkčnosti řídicího programu na jiném operačním systému, který může být základem pro budoucí RT systém nové platformy. Spolu se stávající hardwarovou platformou s mikroprocesorem FUJITSU tedy bude nový řídicí program otestován na třech platformách.

6.1 Návrh GUI nádstavby řídicího programu

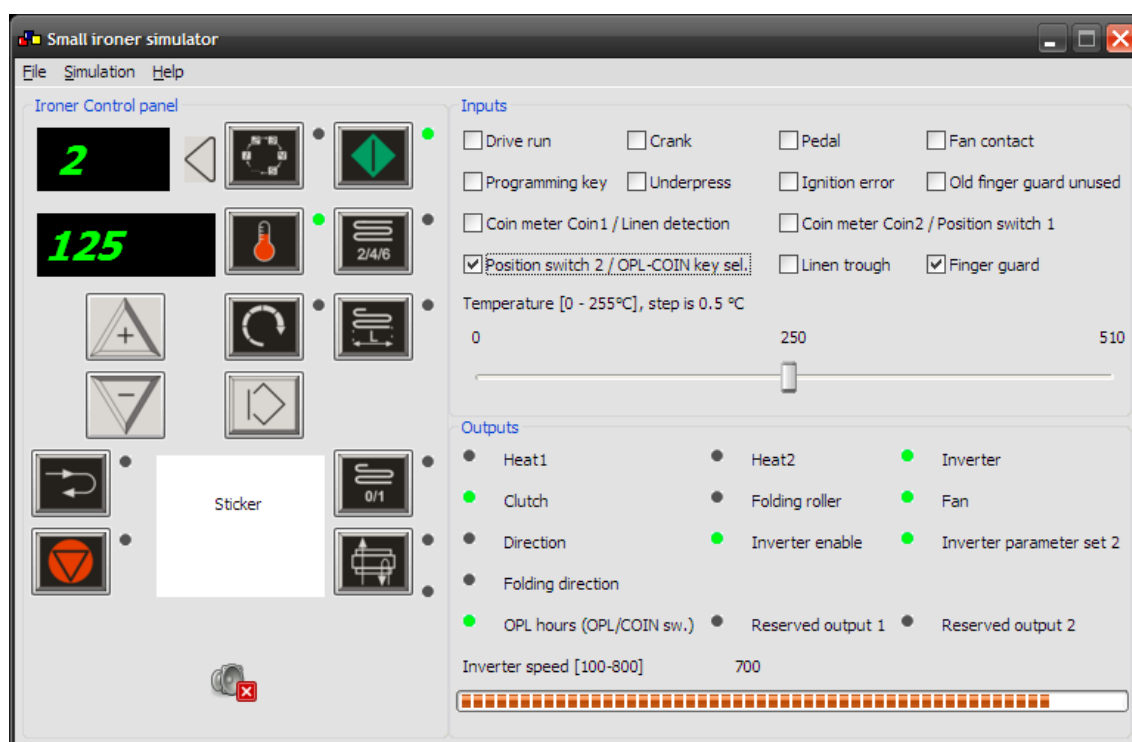
Navržená GUI nádstavba je uvedena na obrázku 14. Levou část navrženého GUI tvoří klávesnice ovládacího panelu stroje a grafická simulace funkce varovné sirénky, pravou pak ovládací prvky pro nastavení logických vstupů, hodnoty analogového vstupu snímání teploty, graficky znázorněné stavy logických výstupů a analogového výstupu řízení frekvenčního měniče.

Samotné provázání GUI s řídicím programem žehliče je provedeno úpravou hlavní funkce řídicího programu. Tato není volána z nekonečné smyčky jako v případě skutečného řídicího programu, ale je cyklicky volána z nově vytvořeného vlákna. Toto vlákno zajišťuje nezávislý běh obsluhy GUI a řídicího programu a provádí synchronizaci stavů vstupů a výstupů mezi GUI a samotným řídicím programem žehliče.

Vzhledem k tomu, že při kompilaci řídicího programu s GUI nádstavbou není možné používat žádné části kódu, které by komunikovaly s vstupními nebo výstupními zařízeními stávající hardwarové platformy, je zajištěno a úspěšnou kompilací i potvrzeno dokonalé oddělení hardwarově závislých částí kódu od samotné logiky řídicího programu. Pro implementaci GUI byla zvolena multiplatformní knihovna wxWidgets [3], vyvíjená pod GPL licenci [4], která umožňuje vytvářet GUI se shodným vzhledem a funkcionalitou na různých platformách na základě stejného zdrojového kódu. Mezi platformy podporované touto knihovnou patří jak Microsoft Windows tak Linux.

6.2 Volba vhodného IDE

Pro efektivní práci na úpravách zdrojového kódu bylo třeba zvolit vhodné vývojové prostředí. Součástí nástrojů dodávaných firmou FUJITSU k mikroprocesoru stávající řídicí desky je kompilátor, linker a jednoduché IDE Softune [5]. Toto vývojové prostředí však již není zřejmě pro stávající platformu nadále vyvíjeno a nedosahuje vlastnostmi a komfortem práce možností moderních IDE.



Obrázek 14: Vzhled GUI nastavy řídicího programu na platformě Microsoft Windows

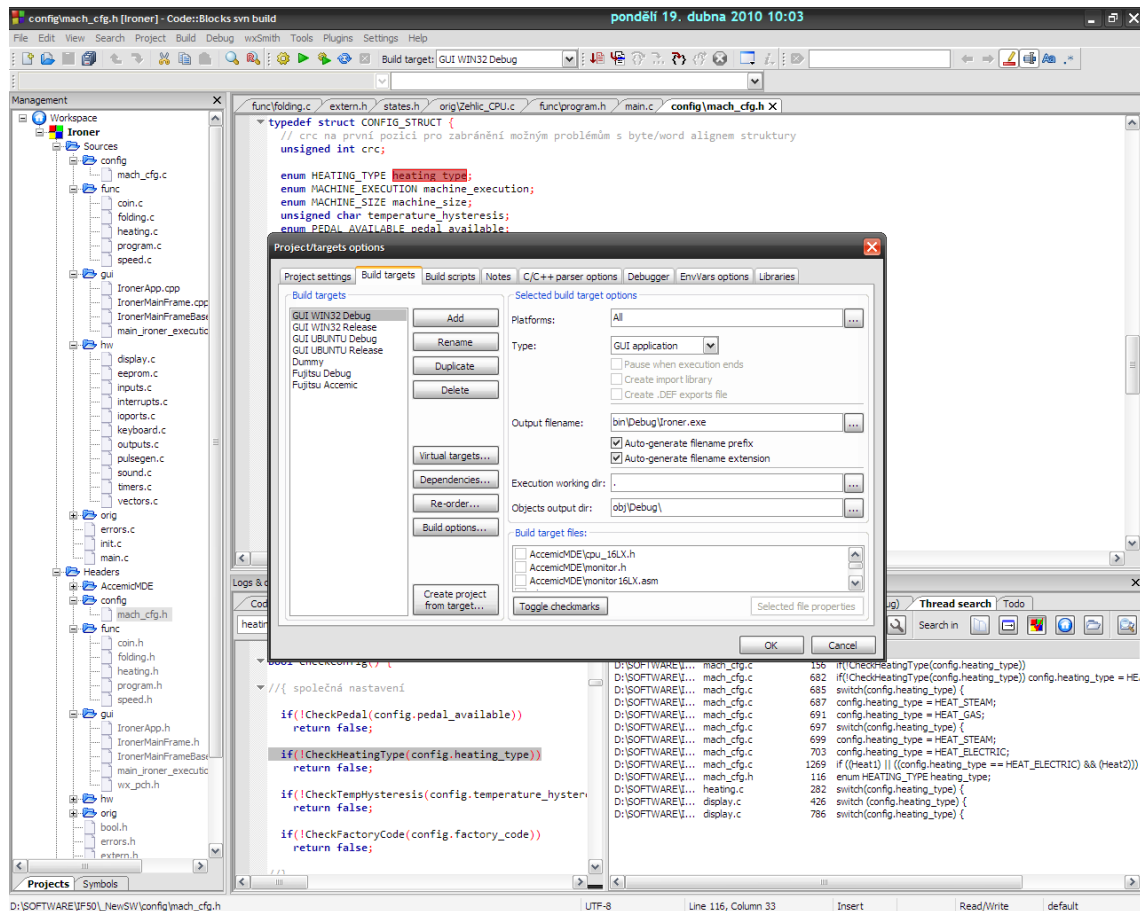
Pro dosažení vyššího komfortu a efektivnosti práce bylo zvoleno jiné IDE a to vývojové prostředí Code::Blocks [6]. Toto IDE umožňuje vývojáři nakonfigurovat si vlastní kompilátor (vedle řady předdefinovaných často používaných kompilátorů) včetně veškerých parametrů příkazové řádky kompilátoru či linkeru. V parametrech projektu pak lze jednoduše přidat vlastní cíl s vlastním kompilátorem a vyvíjet řídicí software na jednu pro několik cílových platform souběžně (viz obrázek 15). Velkou výhodou takto konfigurovaných cílů je i možnost definovat, které ze souborů projektu budou kompilovány a které ignorovány a definování podmínek pro podmíněnou kompilaci. Vyjmutím zdrojových a hlavičkových souborů GUI nádstavby z projektového cíle pro stávající hardwarovou platformu je pak zajištěna jeho nezávislost na navržené GUI nádstavbě, správnou volbou podmínky pro podmíněnou kompilaci je zabezpečena kompilace těch částí zdrojového kódu, které přímo komunikují s hardwarovým vybavením pro stávající hardwarovou platformu.

Dalším velkým přínosem použití moderního IDE a moderních kompilátorů je jejich striktní kontrola správnosti kódu dle normy jazyka C a varování před chybami jako je např. neúplný příkaz `switch` pro proměnnou výčtového typu `enum`, který je hlavním stavebním kamenem pro zpracování funkcí řídicího programu ve formě stavových automatů.

Pro zvýšení efektivity programování je IDE vybaveno také funkcí automatického dokončování klíčových slov, názvů symbolů, proměnných a funkcí a také prohlížečem symbolů pro aktuální soubor nebo celý projekt.

V budoucnu při přechodu na novou hardwarovou platformu je možné opět využít toto IDE přidáním nového kompilátoru, definicí parametrů jeho příkazové řádky, výběrem potřebných zdrojových a hlavičkových souborů a definicí nové podmínky kompilace.

Nemalým kladem tohoto IDE jsou jistě také nulové pořizovací náklady, neboť je vyvíjeno pod GPL licencí [7].



Obrázek 15: IDE Code::Blocks na platformě Microsoft Windows

7 Možnosti portace na RT systém

V průběhu návrhu nového uspořádání řídicí logiky ovládacího programu byl brán zřetel na jeho budoucí převod na RT systém (např. embOS, viz [8]). Proto byly jednotlivé sekce logiky odděleny do zvláštních stavových automatů tak, aby byla zaručena jejich maximální vzájemná nezávislost. Toto oddělení zaručuje v případě nutnosti jednoduchý převod částí řídicí logiky na samostatné úlohy.

Vzhledem k tomu, že úplné nezávislosti všech částí logiky řízení není možné dosáhnout, měly by jednotlivé úlohy být spuštěny ve stejném adresním prostoru (tedy např. jako vlákna nebo jako procesy), aby bylo možno s minimálními změnami kódu zajistit předávání nutných signálů. V dané souvislosti byl proto zvolen oddělený běh logiky ovládání stroje od aktualizace displejů a LED indikátorů stavu. Toto sice přináší jistou redundanci v kódu, naproti tomu může být přínosem po přechodu na novou hardwarovou platformu, která nebude jako zobrazovací prvky displejů využívat sedmi-segmentové zobrazovací prvky, ale plnohodnotného grafického displeje, jehož aktualizace je mnohem náročnější na strojový čas.

Dalším možnou samostatnou úlohou může být obsluha zvukových signálů, která na stávající platformě využívá integrovaného konfigurovatelného generátoru pulzů. V případě nepřítomnosti takového generátoru na budoucí hardwarové platformě bude muset být potřebný pulzní signál generován samostatně, což bude vhodnější provádět ve zvláštní úloze.

Jelikož stávající platforma neumožňuje spouštění více nezávislých úloh, nebyly tyto úlohy definovány a bude nutno jejich oddělení a synchronizaci řešit až dle specifikace a potřeb nové hardwarové platformy.

8 Závěr

Zdrojový kód řídicího programu byl přepracován tak, aby splňoval jak obecné tak firmní požadavky pro strukturovanost a modularizaci zdrojového kódu. Logika řízení byla zpracována a implementována ve formě navržených stavových automatů. Další minoritní úpravy zdrojového kódu a případné rozdělení na samostatné úlohy budou provedeny podle specifikace nové hardwarové platformy a vlastností pro ni zvoleného RT systému.

Nově vzniklý zdrojový kód řídicího programu byl otestován na stávající hardwarové platformě a v součinnosti s navrženou a implementovanou GUI nádstavbou na platformě Microsoft Windows a Linux. Bezchybnou kompilací zdrojového kódu bylo dosaženo ověření dokonalého oddělení hardwarově závislých částí řídicího programu od samotné logiky funkcionality a ovládání stroje. Výsledné řídicí programy byly na uvedených platformách otestovány a byla ověřena stejná funkčnost, jako u stávajícího řídicího programu.

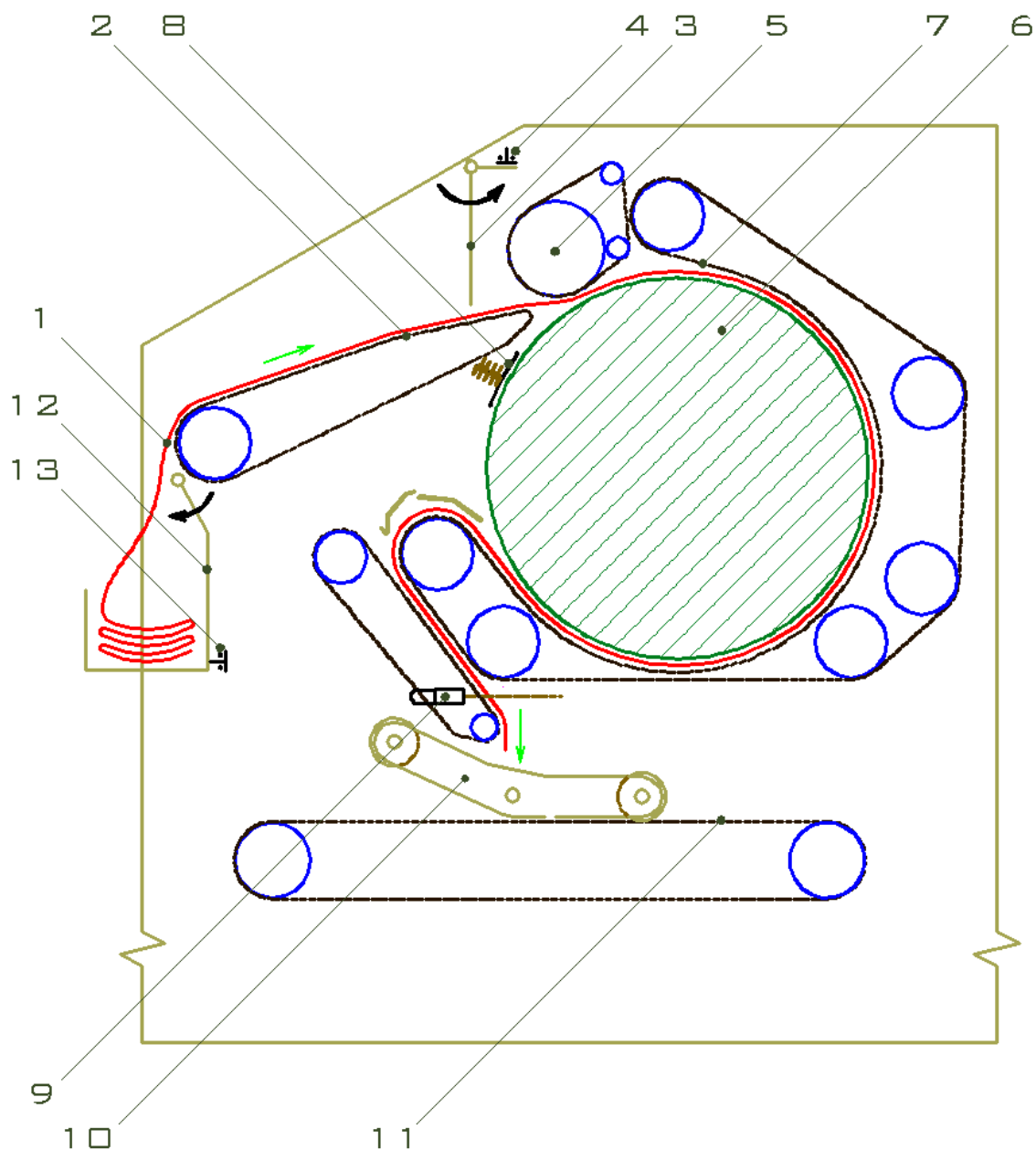
Veškeré poznatky získané v průběhu zpracování diplomové práce, jakož i nemalá část výsledného zdrojového kódu, budou v brzké době využity pro tvorbu řídicího programu nově vyvíjené verze žehliče, který bude dočasně využívat stávající hardwarovou platformu, do budoucna je však také plánována migrace na novou hardwarovou platformu.

Petr Faltýnek

9 Reference

- [1] F2MC-16LX MB90495G Series *FUJITSU MICROELECTRONICS DATA SHEET*
<http://edevice.fujitsu.com/fj/DATASHEET/e-ds/e713713.pdf>
- [2] Programovací manuál *PRŮMYSLOVÉ SUŠICÍ VÁLCOVÉ ŽEHLIČE I35/50, IF50, IR50*
- [3] Cross-Platform GUI library, *wxWidgets* <http://www.wxwidgets.org/>
- [4] *The wxWidgets Licence* <http://www.wxwidgets.org/about/newlicen.htm>
- [5] 16bit Development Environment *F2MC-16 Family SOFTUNE Workbench*
<http://www.fujitsu.com/ca/en/services/edevices/microelectronics/microcontrollers/datalib/softune/flswb16.html>
- [6] The open source, cross platform, free C++ IDE *Code::Blocks*
<http://www.codeblocks.org/>
- [7] *The Code::Blocks IDE Licence* <http://www.codeblocks.org/license/3>
- [8] Real time operating system *embOS* <http://www.segger.com/cms/embos.html>

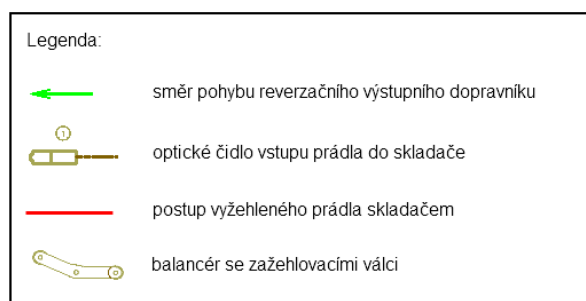
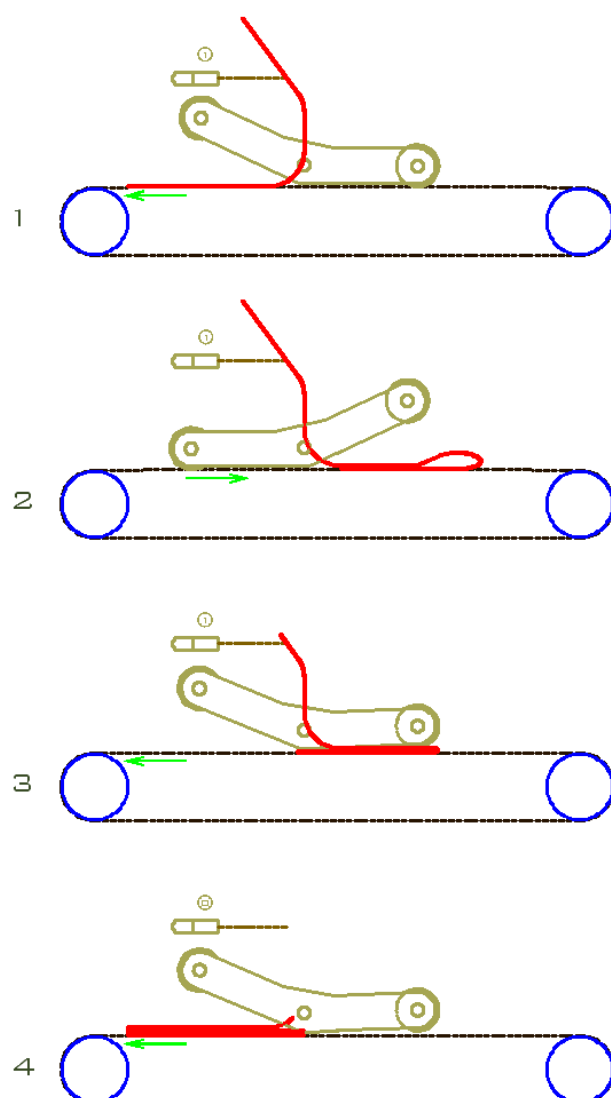
A Schémata a diagramy



Legenda:

- 1 - žehlené prádlo
- 2 - vstupní podávací stůl
- 3 - lišta ochrany prstů
- 4 - senzor stisku lišty ochrany prstů
- 5 - podávací pás
- 6 - vytápěný žehlicí válec
- 7 - žehlicí pásy
- 8 - teplotní senzor
- 9 - optický senzor vstupu prádla do skladače
- 10 - balancér se zažehlovacími pásy
- 11 - výstupní (reverzační) dopravník
- 12 - odklopný žlab
- 13 - senzor odklopení žlabu

Obrázek 16: Schematicky znázorněný princip funkce žehliče



Obrázek 17: Schematicky znázorněný princip funkce skládání prádla